Last Time: + Characterizations of Trees
+ Basic properties

Recall: Every tree w/ at least two vertices has a leaf.

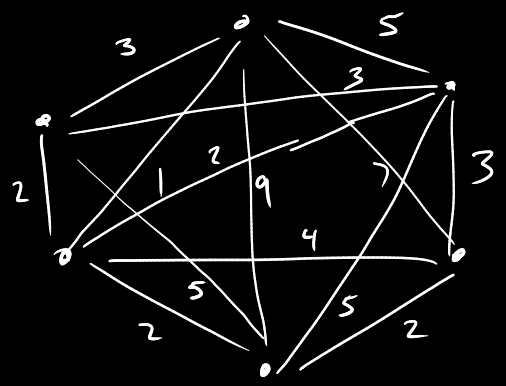
↳ A tree is rooted when we specify a root vertex.

↳ (T, r) is a rooted tree when T is a tree and $r \in V(T)$

Suppose the edges of a graph come with associated costs.

Q: What is the minimum cost of a spanning subgraph?

Ex: Suppose you're an admin for a computer network. You want to minimize the cost of connecting the network. Laying ethernet has costs associated to distances between the servers...



what's the minimum cost?

A: Prim's Algorithm...

Def^n: A weighted graph is a pair (G, w) where G is a graph and $w : E(G) \to \mathbb{R}$.

Problem: Given a connected weighted graph, what is the minimum cost of a spanning tree in G?

Prim's Algorithm: Let (G, w) be a connected weighted graph. To compute a minimum weight spanning tree:

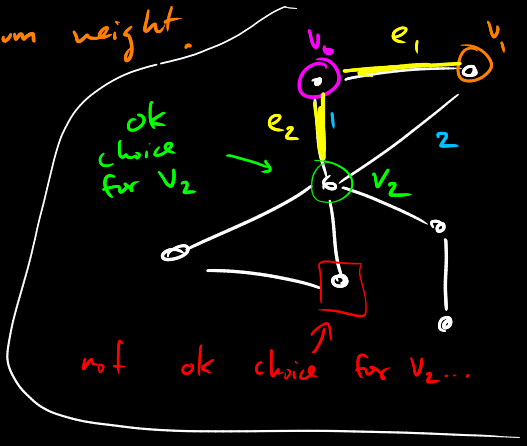
① Choose any vertex v_0 of G. $(T_0 = (\{v_0\}, \emptyset))$

② Suppose we have constructed a tree T_i w/ vertices $v_0, v_1, \dots, v_i$:

ⓐ if $V(G) = \{v_0, v_1, \dots, v_i\}$, Stop.

ⓑ otherwise choose still an edge of minimum weight connecting a new vertex

v_{i+1} to one of $v_0, v_1, \cdots, v_i$ with minimum weight.

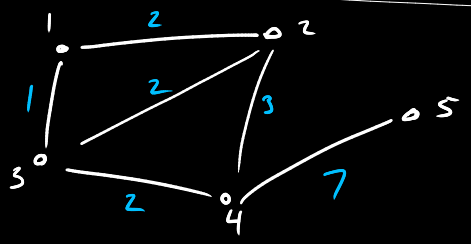


(c) Repeat step 2 with

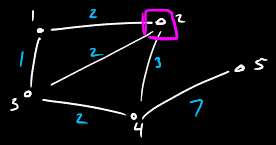
$$T_{i+1} = (\{v_0, \cdots, v_{i+1}\}, \{e_1, e_2, \cdots, e_{i+1}\}).$$

(3) The result of steps (1) and (2) is a tree T_k; Claim: T_k has minimum weight among all spanning trees of G.

Ex: Apply prim's algorithm to (G, w):

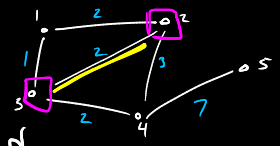


Ssl: First pick a root. $v_0 = 2$

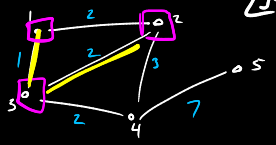


Next(1): Grow the tree by choosing edge 23 of weight 2. (close arbitrarily.)

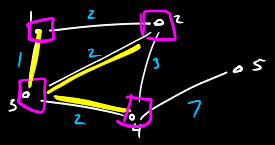
$(e_1 = 23, \quad v_1 = 3)$



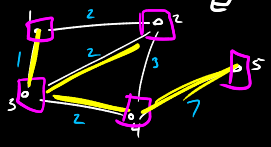
Next(2): Grow the tree by choosing edge $e_2 = 13$, $v_2 = 1$ w/ weight 1. (this choice was forced)



Next(3): Grow the tree by choosing edge $e_3 = 34$, $v_3 = 4$ w/ weight 2.

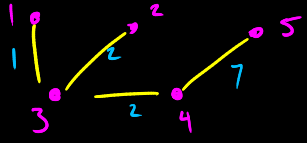


Next(4): Grow the tree by choosing edge $e_4 = 45$, $v_4 = 5$ w/ weight 7.



STOP: $V(G) = \{1,2,3,4,5\} = \{v_0, v_1, v_2, v_3, v_4\}$!!

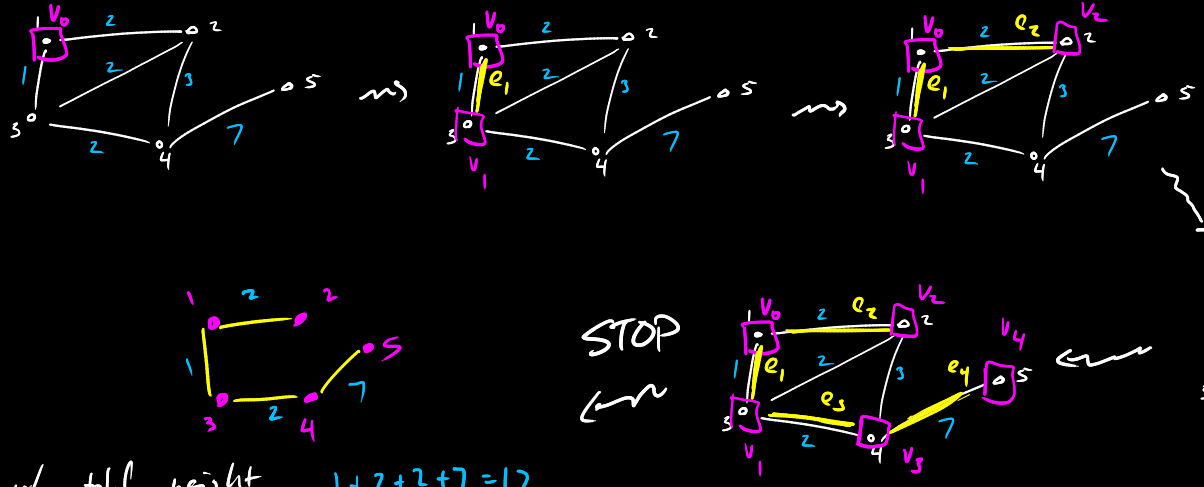
The resulting spanning tree is



of weight $1+2+2+7 = 12$.

NB: it is possible to obtain different spanning trees from this algorithm.

Ex: We make different choices to solve the same problem above...



w/ total weight $1+2+2+7=12$

Observation: We did obtain the same total weight, even with different choices.

Prop: The weight of the output of Prim's algorithm has the minimum possible weight among all spanning trees. pf: Omitted ☹

NB: There are other algorithms to construct min. weight spanning trees...

Kruskal's Algorithm (?): Let (G, w) be a weighted connected graph.

To construct a minimum weight spanning tree:

① $S_0 = \emptyset$.

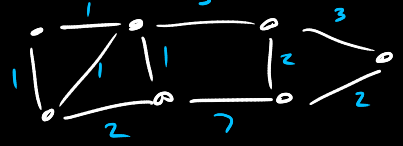
② Given $S_i \subseteq E(G)$, choose an edge $e_{i+1} \in E(G)$ such that
   (a) e_{i+1} does not form a cycle w/ the edges of S_i
   (b) e_{i+1} has the minimum weight among all edges satisfying (a).

   Define $S_{i+1} = S_i \cup \{e_{i+1}\}$.

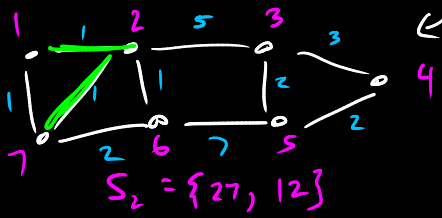
③ Repeat step ② until no longer possible
   (i.e. until every additional edge builds a cycle).

④ Output the final S_K from steps ① - ③.

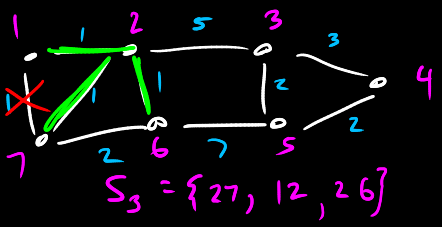
Claim: This S_K is the edge set of a minimum weight spanning tree in G.

Ex: Perform Kruskal's algorithm on  .

Sol:

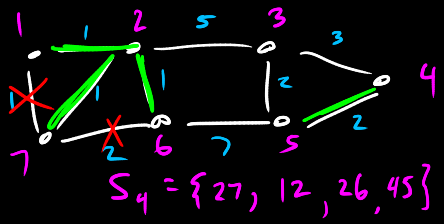


$S_2 = \{27, 12\}$

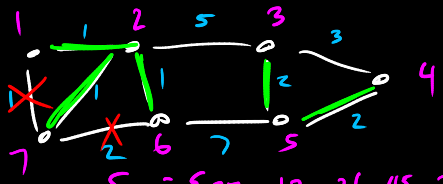


17 would create a cycle :(

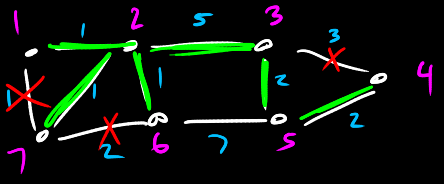
$S_3 = \{27, 12, 26\}$



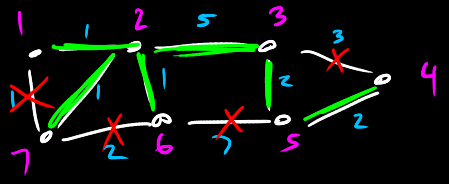
$S_4 = \{27, 12, 26, 45\}$



$S_5 = \{27, 12, 26, 45, 35\}$



$S_6 = \{27, 12, 26, 45, 35, 23\}$

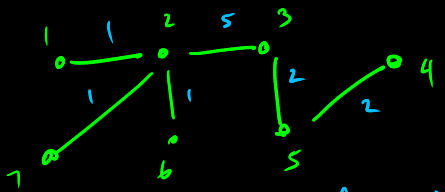


$S_6 = \{27, 12, 26, 45, 35, 23\}$

STOP!

every additional edge would create a cycle.

Output the tree:



has total weight 12.

Ex: Perform both Prim's and Kruskal's algorithm on weighted graph: